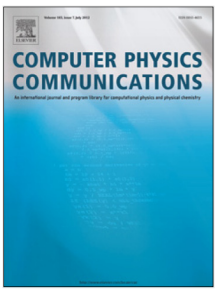
ANTICOOL: simulating positron cooling and annihilation in atomic gases

Green, D. G. (2018). ANTICOOL: simulating positron cooling and annihilation in atomic gases. *Computer Physics Communications*, *224*, 362-370. https://doi.org/10.1016/j.cpc.2017.11.012

Published in:
Computer Physics Communications

Document Version:
Peer reviewed version

Accepted Manuscript



ANTICOOL: Simulating positron cooling and annihilation in atomic gases

D.G. Green

PII: S0010-4655(17)30395-8
DOI: https://doi.org/10.1016/j.cpc.2017.11.012
Reference: COMPHY 6378

To appear in: *Computer Physics Communications*

Received date : 7 July 2017
Revised date : 27 October 2017
Accepted date : 20 November 2017

Please cite this article as: D.G. Green, ANTICOOL: Simulating positron cooling and annihilation in atomic gases, *Computer Physics Communications* (2017), https://doi.org/10.1016/j.cpc.2017.11.012

ANTICOOL: Simulating positron cooling and annihilation in atomic gases ☆

D. G. Green[1]

Centre for Theoretical Atomic, Molecular and Optical Physics, School of Mathematics and Physics, Queen's University Belfast, Belfast BT7 1NN, Northern Ireland, United Kingdom.

Abstract

The Fortran program ANTICOOL, developed to simulate positron cooling and annihilation in atomic gases below the positronium-formation threshold, is presented. Given positron-atom elastic scattering phase shifts, normalised annihilation rates $Z_{\rm eff}$, and γ spectra as a function of momentum k, ANTICOOL enables the calculation of the positron momentum distribution $f(k,t)$ as a function of time t, the time-varying normalised annihilation rate $\bar{Z}_{\rm eff}(t)$, the lifetime spectrum and time-varying annihilation γ spectra. The capability and functionality of the program is demonstrated via a tutorial-style example for positron cooling and annihilation in room temperature helium gas, using accurate scattering and annihilation cross sections and γ spectra calculated using many-body theory as input.

PROGRAM SUMMARY

Program Title: ANTICOOL
Program Files doi: http://dx.doi.org/10.17632/kjstc6xskg.1
Licensing provisions: GPLv3
Programming language: Fortran 90
Nature of problem: Calculation of dynamics of antiparticle cooling and annihilating in atomic gases. Developed for positron cooling and annihilation in atomic gases, but easily modifiable and applicable to other binary elastic-collision cooling processes governed by quantum-mechanical cross sections.
Solution method: Monte Carlo simulation.
Restrictions: Elastic scattering and annihilation only: positron energies must be below the positronium-formation threshold. It is a single particle program, i.e., positron-positron interactions are not included and binary collisions with the gas atoms are assumed.

1. Introduction

Positrons are typically produced in the laboratory at high energy (e.g., ~ 0.5 MeV for traditional ^{22}Na sources, and ~ 1 keV for nuclear reactor sources, e.g., the NEPOMUC reactor [1, 2]). As they propagate through a gas of atoms or molecules they lose energy rapidly through ionization, electronic and rotational excitation, and inelastic processes such as molecular dissociation and positronium formation. At energies below the positronium-formation threshold, energy loss in atomic gases proceeds only via momentum transfer in elastic collisions. Positron annihilation cross sections are typically orders of magnitude smaller than elastic or momentum transfer ones (see, e.g., [3]), and thus positrons tend to undergo many collisions before reaching their inevitable fate of annihilation. Positron cooling dynamics therefore reflect, and thus provide an important probe of, the (complicated) atomic physics of the positron-atom system.

Indeed, the observation of lifetime spectra for positrons annihilating in gases was one of the first sources of information on positron interaction with atoms and molecules (see e.g., [4, 5] for reviews).

☆The associated computer program and corresponding manual will be made available from the CPC library

*Corresponding author. Tel: +44 (0)28 9097 1935.
Email address: d.green@qub.ac.uk (D. G. Green)

Their usefulness as a probe relies on the strong positron-momentum dependence of the annihilation rate and γ spectra: epithermal annihilation manifests in a distinct 'shoulder' of the positron lifetime spectrum and of the related time-varying annihilation rate [4, 5, 6], and an associated distinct 'knee' in the AMOC (Age MOmentum Correlation') spectra [7, 8, 9, 10, 11, 12], where the positron 'age' (i.e., lifetime from source to annihilation) is measured in coincidence with the emitted γ rays.

Understanding positron cooling or moderation via buffer gases is important to develop positron traps and accumulators [13], used in e.g., the antihydrogen experiments at CERN, and to develop an ultra-high-resolution cryogenic positron beam [14, 15], which is required for the study of positron-induced intramolecular vibrational redistribution and to make state-resolved measurements of rotational excitation and vibrational multimode excitation in molecules.

The fundamental dynamics of positron cooling in atomic and molecular gases is governed by the Fokker-Planck equation [16]. It has been the method of choice for the majority of the handful of previous theoretical calculations of positron cooling in noble gases [17, 18, 19, 20, 21, 22], which have typically relied on model scattering and annihilation cross sections, and have yielded limited agreement with experiment. As first demonstrated by Farazdel and Epstein [23], a powerful and versatile alternative approach to the study of positron cooling in atomic and molecular gases is offered by Monte Carlo (MC) simulation. Recently, MC calculations based on accurate many-body-theory calculated scattering and annihilation cross sections have been used to elucidate the dynamics of positron cooling and annihilation in noble gases [6, 12]. Those calculations showed that the number of positrons surviving to thermalization was strikingly small, and particle loss due to annihilation effected the time-varying annihilation rate $Z_{\rm eff}(\tau)$. The best agreement with experiment to date was found for all the noble gases, except for Ne, where it was proferred that the experiment suffered from incorrect analysis and/or the presence of impurities.

This paper presents the `ANTICOOL` MC program[1] on which those calculations were based. It is aimed at experimentalists and theoreticians whose aim is to understand the dynamics of positron (or electron) cooling in atomic gases. Specifically, given user supplied scattering phase shifts, annihilation rates and annihilation γ spectra for the positron-atom system of interest, `ANTICOOL` enables the calculation of the time evolution of the positron momentum distribution $f(k,t)$, the fraction of positrons annihilating in a given time (lifetime spectrum), and the time-varying annihilation rate and γ spectra for positrons cooling in a thermal atomic gas[2]. These quantities can be measured in positron lifetime [4, 5] or AMOC experiments [7, 8, 9, 10, 11].

The program was developed and tested to study positron cooling in noble gases, using high-quality positron-momentum-dependent elastic scattering phase shifts, normalised annihilation rates $Z_{\rm eff}$ and annihilation γ spectra calculated using many-body theory. It was also found to produce results that were in excellent agreement with complementary calculations that directly solved the Fokker-Planck equation numerically [24].

The structure of the remainder of the paper is as follows. Section 2 describes the details of the Monte Carlo algorithm employed by `ANTICOOL` to simulate positron cooling and annihilation in a thermal atomic gas. Section 3 demonstrates the basic functionality of the program (and associated post-processing programs) via a tutorial style example of positron cooling in room-temperature helium gas [25] (see the `.../anticool/example_He/` directory). The paper concludes with a brief summary and outlook.

[1] `ANTICOOL` is written in modular serial Fortran and can be run on a modest desktop computer.

[2] It is assumed that the positron energy is below the positronium-formation threshold, corresponding to the energy range probed in experiments [4, 5].

2. Simulation of positron cooling and annihilation in atomic gases as implemented in ANTICOOL

2.1. Determination of the time-evolving positron-momentum distribution

The program determines the time-evolving momentum distribution for positrons cooling in the gas $f(k,\tau)$[3], normalised as $\int_0^\infty f(k,\tau)dk = F(\tau)$, the fraction of initial positrons remaining, as follows. The momentum $k(\tau_i)$ of an individual positron is determined over an equidistant grid in time-density $\{\tau_i\}$ with step size $\Delta\tau$ as follows (see also Figs. 1 and 2 for schematics of the program structure).

At each time step the lab-frame velocity of one gas atom is sampled from the Maxwell-Bolztmann distribution at the user specified temperature. Specifically, the x, y and z components of the velocity are sampled from their respective (Gaussian) distributions, implemented via the Box-Muller transformation, i.e., each component is determined as

$$v_i = \sqrt{\frac{k_B T}{M}}\mathcal{N}(0,1) \;\; ; \;\; i = x,y,z, \tag{1}$$

where M is the mass of the gas atom and $\mathcal{N}(0,1)$ is a Gaussian-distributed random number with zero mean and unit standard deviation, so that the speed $v = \sqrt{\sum_i v_i^2}$ is Maxwell-Boltzmann distributed as required.

The positron and the gas-atom velocities are transformed to the centre-of-mass (COM) frame. The relative velocity

$$\mathbf{v}_r = \mathbf{v}_p - \mathbf{v}_g = \mathbf{u}_p - \mathbf{u}_g, \tag{2}$$

and relative speed $v_r = |\mathbf{v}_r|$ of the positron and gas atom are then calculated. Here $\mathbf{v}$ are the lab-frame velocities, and $\mathbf{u}$ are the COM-frame velocities of the particle (subscript 'p') or gas atom (subscript 'g') which are related as

$$\mathbf{u}_{p,g} = \mathbf{v}_{p,g} - \mathbf{V}_{\rm CM}, \tag{3}$$

where the COM velocity

$$\mathbf{V}_{\rm CM} = \frac{\sum_{i=p,g} m_i \mathbf{v}_i}{\sum_i m_i} = \frac{m\mathbf{v}_p + M\mathbf{v}_g}{m+M}. \tag{4}$$

The energy available for the collision in the COM frame is $E_{\rm CM} = \mu v_r^2/2$, where μ is the reduced mass of the positron and gas atom. Next, a uniformly distributed random number $r_1 = U[0,1]$, is drawn, and a collision is deemed to occur if $r_1 < P = W\Delta\tau$, where $W = n_g v_r \sigma_{\rm tot}$ is the rate of a collision event (either annihilation or elastic scattering), with $\sigma_{\rm tot} = (\sigma_{\rm el} + \sigma_{\rm a})$, subject to the requirement that $P = W\Delta\tau \ll 1$ (in practice we demand that $P = W\Delta\tau < 0.1$)[4]. Here $\sigma_{\rm el}$ is the positron-atom elastic scattering cross section and $\sigma_{\rm a}$ the cross section for positron annihilation with the atomic electrons. The elastic cross section is determined as the integral of the differential elastic cross section [26]

$$\varrho = \frac{d\sigma_{\rm el}}{d\Omega} = |f(\theta)|^2, \tag{5}$$

where $f(\theta)$ is the scattering amplitude,

$$f(\theta) = \sum_{l=0}^{\infty}(2l+1)\frac{e^{2i\delta_l}-1}{2ik}P_l(\cos\theta)\,, \tag{6}$$

for scattering angle θ, and δ_ℓ the elastic scattering phase shifts. The elastic scattering phase shifts must be supplied as input. For the example test case that we will consider below, we calculate the scattering amplitude using accurate phase shifts for positron angular momenta ℓ =0, 1 and 2 calculated using many-body theory [25], and approximate the phase shifts for all $\ell > 2$ well using the leading k^2 term in the expansion [27] [see Sec. III D of [25] for details, specifically Eqn. (31)][5]. The cross

[3] We work in units where τ is the time (in ns) scaled by the number density of the gas n_g (in amagat): $\tau = n_g t$.

[4] When setting up a new simulation, the user should first run the code for a single particle to ensure that $W\Delta\tau$ is below 0.1: see the screen output `Max(Prob_event)`, which should be less than 0.1, otherwise an explicit error message is produced.

[5] For $\ell > 2$ one can also choose to include the $O(k^4)$ corrections of Ali and Fraser [28], although these corrections have negligible effect on the cooling process for positrons in room-temperature noble gases.

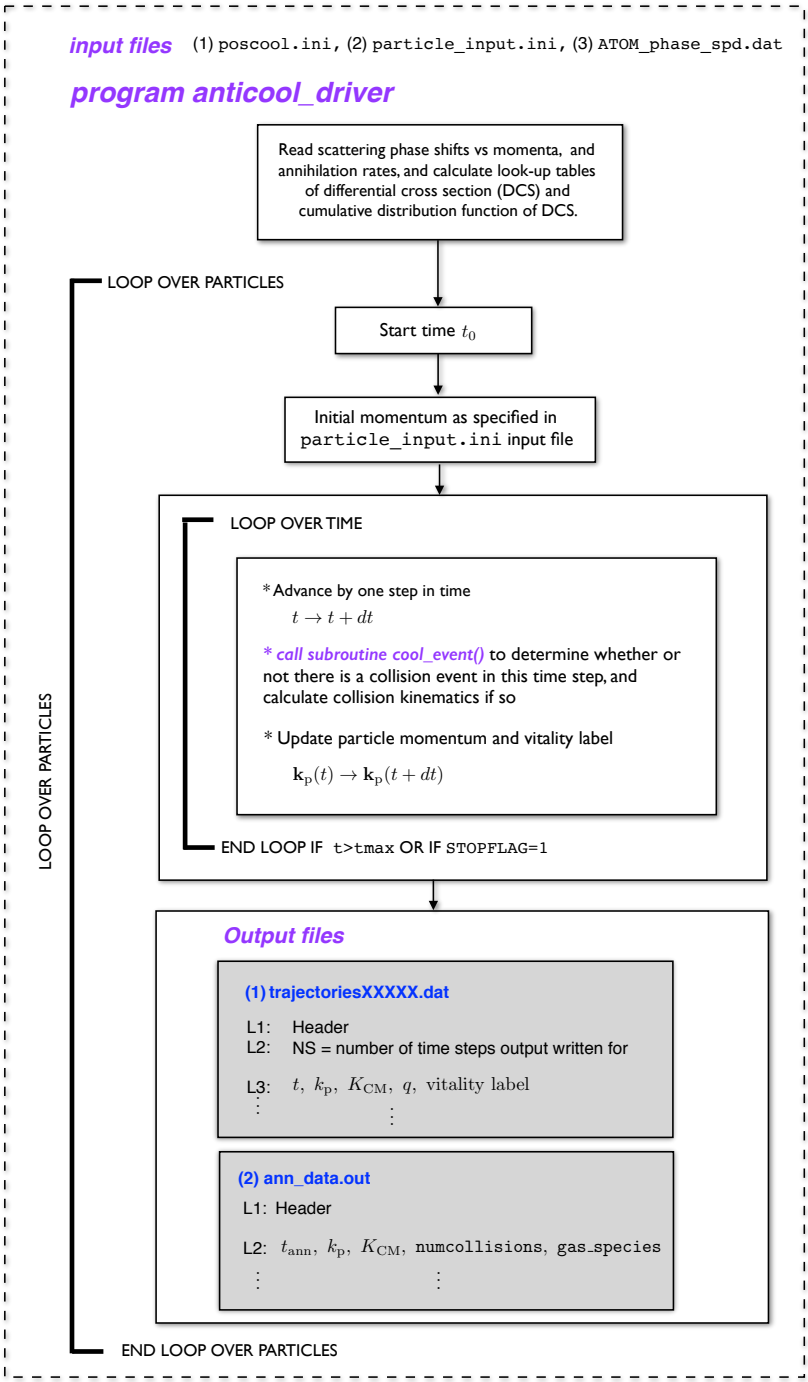


Figure 1: Basic structure of the ANTICOOL program, showing input and format of main output files.

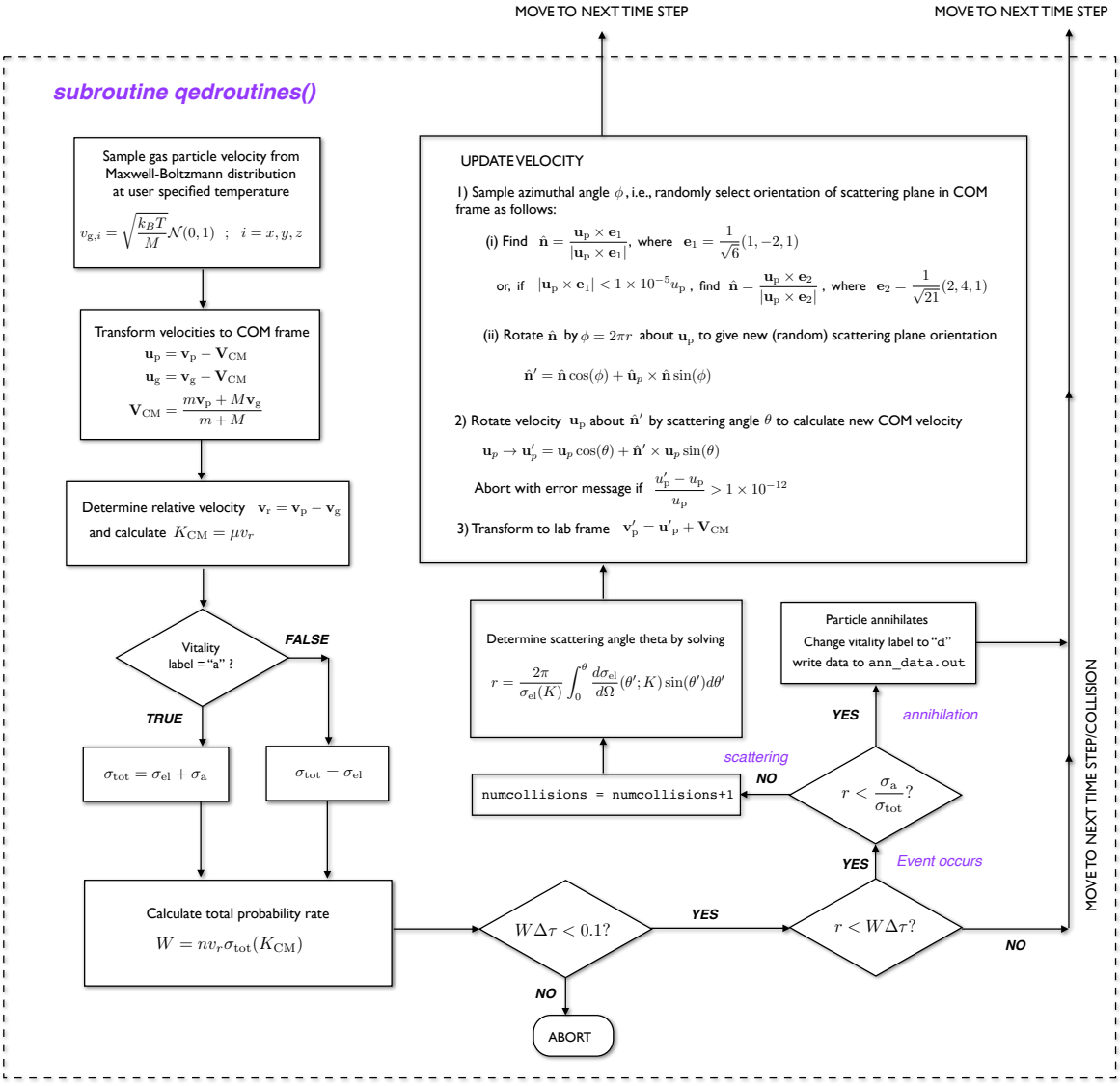


Figure 2: Structure of the collision event generator, which updated the particle velocity and 'vitality' label.

section for positron annihilation in a many-electron target atom is [29, 30]

$$\sigma_a = \pi r_0^2 \frac{c}{v_r} Z_{\rm eff}. \tag{7}$$

Here r_0 is the classical electron radius, c is the speed of light, v_r is the relative velocity of the positron and target atom, and $Z_{\rm eff}(k)$ is the *effective* number of electrons that contribute to the annihilation, or so-called *normalised annihilation rate*. It is normally parameterised as $Z_{\rm eff} = \lambda/\lambda_D$, the ratio of the true annihilation rate λ to the 'Dirac' rate $\lambda_D = \pi r_0^2 c n_g$ of free positron annihilation in a gas of number density n_g. Positron-electron and positron-atom correlations can make $Z_{\rm eff}$ much greater than the number of valence electrons with which the positron predominantly annihilates [25, 31].

If a collision event is deemed to occur, it must be determined whether it is an elastic scattering or annihilation. Thus, a second random number $r_2 = U[0,1]$ is drawn. If $r_2 < \sigma_a/\sigma_{\rm tot}$ the event is deemed to be annihilation and the particle is removed from the simulation, otherwise it is an elastic collision and the positron velocity is updated as follows. The scattering angle θ is sampled from the differential cross section by finding the root of

$$r_3 = \frac{2\pi}{\sigma_{\rm el}} \int_0^\theta \varrho \sin\theta' d\theta', \tag{8}$$

where $r_3 = U[0,1]$. In the centre-of-mass frame elastic scattering is symmetric with respect to the azimuthal angle ϕ, which is thus chosen randomly.

Trajectory data for each positron in the simulation is printed at each time step to the output file '`trajectoriesXXXXX.out`', suffixed by the positron number. If a positron annihilates during the specified simulation time, then the time of annihilation, and positron momentum at that time is added to the output file '`ann_data.out`'. The post-processing program '`collater.f90`' reads these files (as described below), and determines the momentum distribution $f(k,\tau)$ by binning the positron momenta at each τ_i. It also produces the data for plotting the time-varying annihilation rate $Z_{\rm eff}(\tau)$ and lifetime spectra.

2.1.1. *Calculation of the time-varing normalised annihilation rate* $Z_{\rm eff}(\tau)$

Knowledge of the positron-momentum distribution enables calculation of the time-varying normalised annihilation rate

$$\bar{Z}_{\rm eff}(\tau) = \int_0^\infty Z_{\rm eff}(k) f(k,\tau) dk. \tag{9}$$

which can be compared with experiment [4, 5, 6]. Epithermal annihilation results in a distinct 'shoulder' in $\bar{Z}_{\rm eff}(\tau)$ [4, 5].

2.1.2. *Calculation of the time-varing* γ *spectra*

In the dominant process, a positron of momentum $\mathbf{k}$ and energy $\varepsilon = k^2/2$ annihilates with an atomic electron in state n to form two γ-ray photons of total momentum $\mathbf{P}$ [32]. In the center-of-mass frame the two γ rays have equal energies $mc^2 = 511$ keV (neglecting the initial positron and electron energies). In the laboratory frame the photon energies are Doppler shifted by $\epsilon \leq Pc/2$, giving rise to a spectrum $w_k(\epsilon)$ that is characteristic of the atom involved and also of the positron momentum at the instant of annihilation [33, 31, 12]

Knowledge of the positron-momentum distribution also enables calculation of the *time-varying* γ spectrum produced by positrons cooling in gases, viz.,

$$\bar{w}_\tau(\epsilon) = \int_0^\infty f(k,\tau) w_k(\epsilon) dk. \tag{10}$$

The characteristic trajectory followed by the positrons in (k,τ) space, along with the dependence of the γ spectra on the positron momentum, leads to a characteristic AMOC spectrum, i.e., the number of γ rays $\tilde{N}_\gamma$ (per unit positron) detected per unit time and Doppler-shifted energy. It can be measured in experiments [7, 8, 9, 10, 11, 34] and calculated as

$$\frac{d^2\tilde{N}_\gamma}{d\tau d\epsilon} = 2\pi r_0^2 c F(\tau) \bar{w}_\tau(\epsilon). \tag{11}$$

The AMOC spectra exhibit distinct 'knee' due to epithermal annihilation (see below and [12]).

2.2. Positron lifetime spectrum

Integrating over the Doppler-shifted energy ϵ gives the lifetime spectrum (normalised to one positron)

$$A(\tau) = d\tilde{N}_\gamma/2d\tau, \quad (12)$$
$$= \pi r_0^2 c F(\tau)\bar{Z}_{\rm eff}(\tau). \quad (13)$$

Epithermal annihilation results in a distinct 'shoulder' of the lifetime spectrum [4, 5].

2.3. γ spectrum $\bar{S}$ and $\bar{W}$ parameters

Once the time-varying γ spectra or AMOC spectra have been determined, the dimensionless time-varying γ-spectra shape parameters

$$\bar{W}(\tau) \equiv 2\bar{Z}_{\rm eff}(\tau)^{-1} \int_{\epsilon_W}^{\infty} \bar{w}_\tau(\epsilon) d\epsilon, \quad (14)$$

and

$$\bar{S}(\tau) \equiv 2\bar{Z}_{\rm eff}(\tau)^{-1} \int_{0}^{\epsilon_S} \bar{w}_\tau(\epsilon) d\epsilon, \quad (15)$$

can be determined (in `ANTICOOL` the limits are hard coded as ϵ_W = 2.0 keV and ϵ_S=0.5 keV) [12]. $\bar{S}(k)$ and $\bar{W}(k)$ respectively characterise the positron momentum dependency of the low and high-Doppler shift energy part of the γ spectrum. They provide a sensitive probe of the positron cooling dynamics (see e.g., [7, 34, 6]).

3. Tutorial style example: positron cooling in room-temperature helium gas

To demonstrate the capability and functionality of the code, this section presents a tutorial-style example of positron cooling in room temperature helium gas. Specifically, we consider simulation of 10,000 positrons cooling in He gas of temperature T = 293 K, for positrons initially distributed uniformly in energy up to the Ps-formation threshold.

3.1. Initial setup of input files

The program `ANTICOOL` is controlled by two input files:

1. "`poscool.ini`", which contains the namelist

```
&SIMPRMS
tmin = 0.d0,
tmax = 3000.d0,
initialdt = 5d-4,
writeevery = 6d4,
QEDannih = .TRUE.,
&
```

that specifies the time-density over which to run the simulation via the start and end times `tmin` and `tmax`, the time step size $\Delta\tau$ via the variable `initialdt` (all specified in units of ns amg), and the number of timesteps between which the trajectory data is written to file. In this case there will be a total of 6,000,000 time steps, and data will be written for a total of 100 times equidistant between `tmin`=0 and `tmax`=3000 ns amg. The `QEDannih` variable specifies whether an inelastic channel should be included: if the particle species is chosen to be the positron (see below), then `QEDannih=.TRUE.` means that annihilation channel will be open (i.e., annihilation will be included).

This input file also contains the namelist

```
&GASPRMS
tempkelvin=293.d0,
atom="he",
phasefile="He_spd_ph.dat"
&
```

The first variable in this namelist specifies the temperature (in Kelvin) and the second the species of the gas via a two letter (lowercase) atomic symbol[6]. The third variable specifies the input file containing the scattering phase shifts that must be supplied by the user. Its

[6] The current version accepts the noble gases `"he"`, `"ne"`, `"ar"`, `"kr"` and `"xe"`. Calculations can be performed by setting `atom="at"` and adding the parameter `"m_at = X"` to the namelist `GASPRMS`, where `X` should be the mass of the atom in amu. Moreover, for atoms other than the aforementioned noble gases, the user must add the Padé-approximant form of the annihilation rate $Z_{\rm eff}(k)$ (see below) if `QEDannih=.TRUE.`

format is discussed in the Appendix (see Table A.3).

2. "`particle_input.ini`".

   This file contains the data describing the initial parameters for the particle (in this case positron). This file is different to the first in that its ordering is important. For this example it should be set as follows:

```
! ...comments
repeatfirstline=on
1  ! number of particle runs
1  ! switch for energy distribution
1, p, 1.09d0
! ...comments
```

   The first line of this file is a header for comments. The second line can be set to `repeatfirstline=off` or `repeatfirstline=on`, depending on whether the user wishes to run simulations using successive lines of distinct specified particle data in turn (thus enabling the user to specify any initial parameter distribution) or instead wishes to run the simulation for multiple particles using the same initial parameters, or according to a distribution that is uniform in energy. We set it to '`on`' as we want an initial distribution that is uniform in energy (see below). The third line must contain an integer specifying the number of runs (i.e., particles to simulate). Let us first run the program for a single positron, so we set this to `1d0`. Setting the fourth line to '`1`', with `repeatfirstline=on`, specifies that an initial uniform *energy* distribution of positrons should be used, with energy from 0 up to $E_{\rm max} = k_{\rm max}^2/2$, where $k_{\rm max}$ is the (upper limit of) the positron *momentum* specified in the final line, here set as 1.09 a.u. (alternatively, setting the fourth line to '`0`' specifies that a monoenergetic initial distribution is used, at the energy $E_{\rm max}$). Each subsequent line of the file specifies the particle label (1 in this case), the particle species: electron (e) or positron (p); and the particle momentum upper limiter for each particle in turn. (If the species is set to '`e`', (electron) then annihilation is turned off, i.e., the program defaults to `QEDannih=.FALSE.`).

3.2. *Compiling and running the code*

Compilation of the code is facilitated via the Makefile (it assumes either an Intel Fortran compiler or the gfortran compiler on a computer with either `x86_64` or `i386` architecture: the user should modify the specified architecture or compiler Makefile variables if alternative compilers or architectures are to be used). Simply navigate to the directory of the source files '`.../anticool/src/`' and type '`make clean`', followed by '`make anticool`'. This should produce an executable in the folder `.../src/x86_64/` (or `.../src/i386/` depending on the computer architecture)[7].

To run the code, one should first create a directory that contains the scattering phase shift (and $Z_{\rm eff}$, if not using the built-in Padé-approximant forms for the noble gases) data files, and both the '`poscool.ini`' and '`particle_input.ini`' input files. If the `x86_64/` (or `.../src/i386/`) directory was added to the users path, the executable can be run from the directory containing the input files simply by typing '`anticool`'. Otherwise, one should copy the executable from '`.../src/x86_64/`' (or `.../src/i386/`) to the data directory before running it by typing '`./anticool`'.

3.3. *Post processing*

Running '`./anticool`' creates a number of output files in addition to the '`trajectories*.out`' and '`ann_data.out`' files. From the input elastic scattering phase shifts, `ANTICOOL` calculates the elastic scattering and momentum-transfer cross sections as a function of momentum, which are output to the files '`esxsec.out`'

[7] It is convenient to add the `.../anticool/src/x86_64` (or `.../src/i386/`) directory to the users path.

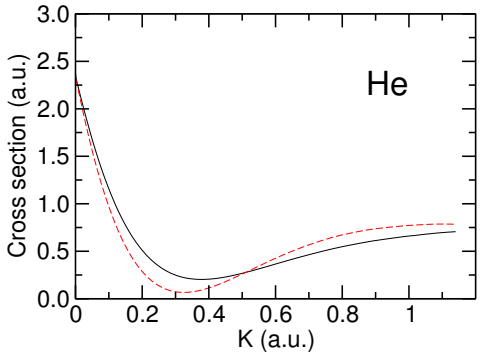


Figure 3: Elastic scattering (solid black line) and momentum-transfer (red dashed line) cross section for positron-helium scattering calculated in the program and output to the files `esxsec.out` and `mtxsec.out`

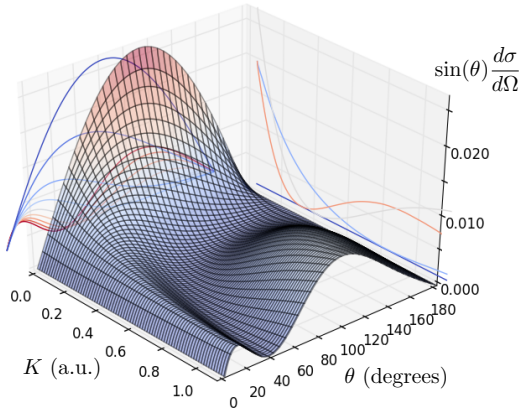


Figure 4: $\sin(\theta)\varrho$, where ϱ is the differential cross section for positron-helium scattering calculated in the program (plot produced using python script `dcs_3d.py`.)

and '`mtxsec.out`' respectively (see Fig. 3). The differential cross section $\varrho(k,\theta)$ [more precisely, $\sin(\theta)\varrho(k,\theta)$] can be plotted as a function of k and θ by running the python script '`dcs_3d.py`' (simply type '`python dcs_3d.py`'[8]), which reads the files '`ak_3d.out`', '`theta_3d.out`' and '`dcs_3d.out`' (see Fig. 4)

For each particle in the simulation run, the `trajectories*.out` file contains (after two header lines) a table of the simulation time-density τ, the positron momentum k at that time, and the

[8]To run succesfully one requires an installation of python and the python "matplotlib" library: see https://matplotlib.org/users/installing.html

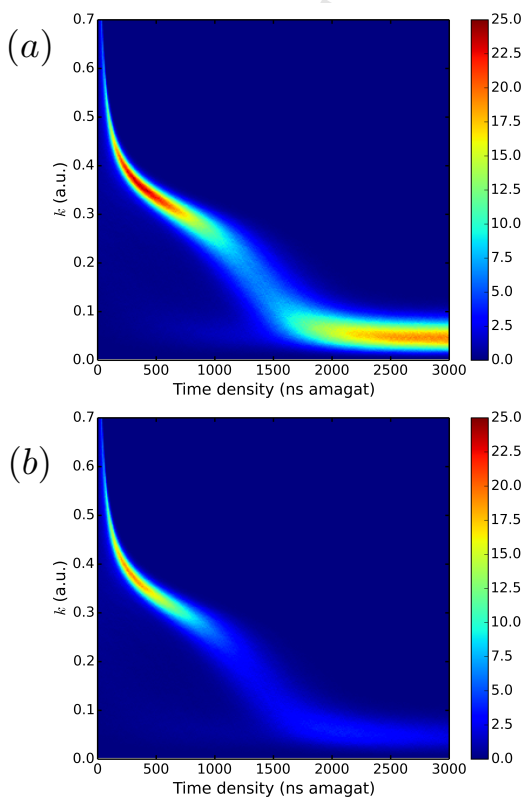


Figure 5: Density plot of positron-momentum distribution $f(k,\tau)$ for He: (a) excluding depletion of the distribution due to annihilation, with $f(k,\tau)$ normalised to unity, and; (b) including depletion of the distribution due to annihilation, normalised as $\int_0^\infty f(k,\tau)dk = F(\tau)$, the fraction of positrons surviving (dashed-dotted line). Results of a simulation involving 50,000 positrons initially distributed uniformly in energy up to 1.1 a.u. (close to the Ps-formation threshold). Such plots can easily be produced by running the postprocessing program `collater.f90` and the python script '`fk2d.py`' (the user must specify the filename to be read by this python script). Also note that the distribution $f(k,t)$ at each time is stored in the output files '`kdist.out`' (excluding depletion) and '`kdist_alive.out`' (including depletion due to annihilation): the first column of these files gives the momentum, and the subsequent columns $f(k,\tau_i)$ for all τ_i. From this, videos of the time-evolving positron momentum distribution can be readily made [35].

'vitality label' of the positron, which is `"a"` (alive) if the positron has not yet annihilated, and `"d"` (dead) if it has annihilated before that corresponding time (see Table 1). The '`ann_data.out`' file contains the time-density τ at which each positron in the simulation annihilated, its momentum at the

Table 1: Format of `trajectories*.out` files

L1	# Comments header		
L2	No. times k is output for		
L3	τ_1 (ns amg)	k_1 (a.u.)	a
⋮	⋮	⋮	
LN	τ_N (ns amg)	k_N (a.u.)	d

instant of annihilation, and the number of elastic collisions it underwent before annihilating (see Table 2)

Table 2: Format of `ann_data.out` file

L1	# Header		
L2	τ_1 (ns amg)	k (a.u.)	No. collisions
⋮	⋮	⋮	
LN	τ_N (ns amg)	k (a.u.)	No collisions

Before moving on, we should run the simulation for 1000 positrons to enable generation of statistical quantities. Thus, set the second line in '`poscool.ini`' to 1000 and re-run the `anticool` executable (such a run takes ∼ 45 minutes on my modest desktop). Once the `anticool` program has completed, the '`trajectories*.out`' and '`ann_data.out`' files can be processed to generate the positron momentum distribution $f(k,\tau)$, the time-varying annihilation rate $Z_{\rm eff}(\tau)$ [Eqn. 9], as well as the AMOC- [Eqn. 11] and positron lifetime spectra Eqn. 12]. This is achieved through the post-processing program '`collater.f90`', which is controlled by the input file '`collater.ini`'. It contains the namelist

```
&COLL
atom="he",
tempk=293.d0,
numparticles=1000,
&
```

which specifies the gas species and its temperature, the total number of particles to read (i.e., the total number of '`trajectories*.out`' files to collate. It also contains a second namelist:

```
&GAMSPECIN
```

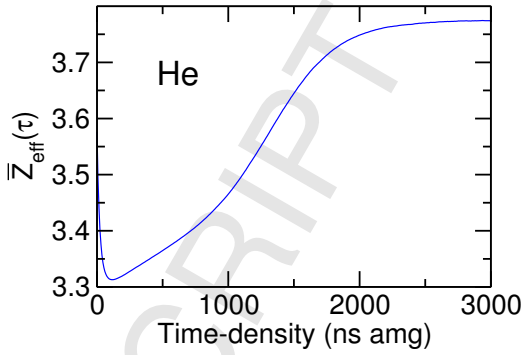


Figure 6: The time-varying annihilation rate $\bar{Z}_{\rm eff}(\tau)$ for positrons cooling in room-temperature helium [plotted using column 1 and column 3 (i.e., depletion due to annihilation included) of the output file '`zefft.coll`'].

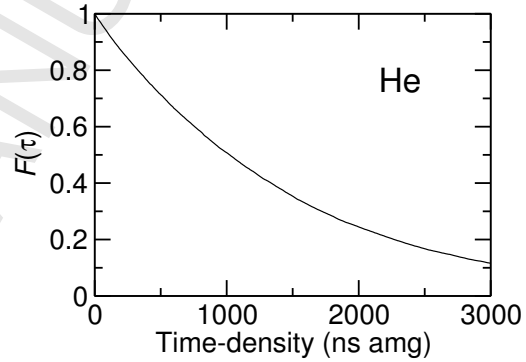


Figure 7: Fraction of positrons surviving at time τ for positrons cooling in room-temperature helium (plotted using data written to output file '`ann_frac.coll`').

```
plotgamspec=.TRUE.,
gamspecfile="gamspec_he.dat",
maxtime=3000.d0,
&
```

This post-processing program is run by typing '`coll`'.

Figure 6 shows the calculated time-varying annihilation rate $\bar{Z}_{\rm eff}(\tau)$ [Eqn. (9), plotted using column 1 and column 3 (i.e., depletion due to annihilation included) of the output file '`zefft.coll`'], and Fig. 7 the fraction of positrons surviving at time-density τ (plotted using the data in the '`ann_data.coll`' output file). Knowledge of both $Z_{\rm eff}(\tau)$ and $F(\tau)$ enables the positron-lifetime spectrum to be calculated via Eqn. (12) (see Fig. 8).

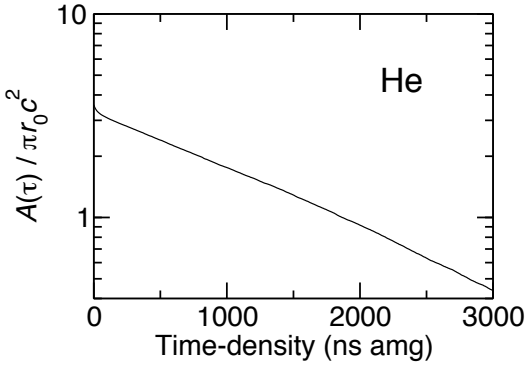


Figure 8: Lifetime spectrum for positrons cooling in room-temperature helium [Eqn. 12].

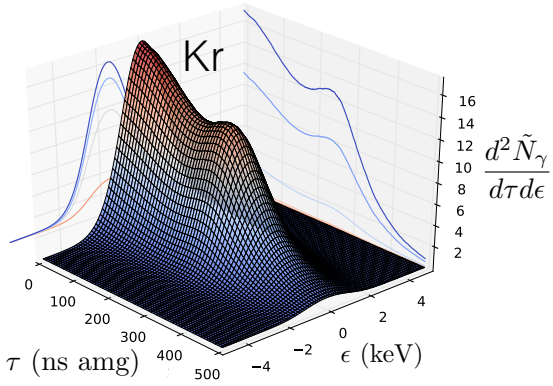


Figure 9: AMOC spectrum [Eqn. 11] for positrons annihilating in krypton, calculated using positrons initially distributed uniformly in energy.

The AMOC spectra [Eqn. 11] can also be plotted, using the python script `amoc.py` (simply type '`python amoc.py`'). To calculate and plot the $\bar{S}(\tau)$ and $\bar{W}(\tau)$ parameters, after the execution of `coll` has completed, one should run the executable '`amoc_sw`' As an example of a 'production' run, Fig. 9 shows the AMOC spectrum calculated using 50,000 positrons cooling in room-temperature *krypton*, and Figs. 10 and 11 the associated $\bar{S}(\tau)$ and $\bar{W}(\tau)$ parameters (plots made from the data file `swt.dat`'.

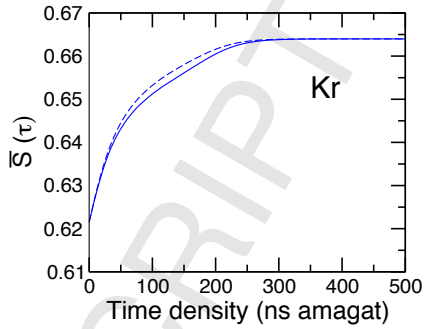


Figure 10: $\bar{S}(\tau)$ [Eqn. 11] for positrons annihilating in krypton, calculated using positrons initially distributed uniformly in energy, excluding and including depletion of the distribution due to annihilation (dashed and solid lines, respectively) .

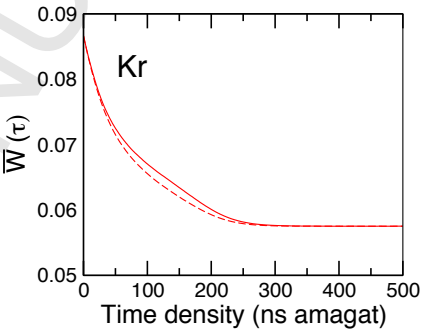


Figure 11: $\bar{W}(\tau)$ [Eqn. 11] for positrons annihilating in krypton, calculated using positrons initially distributed uniformly in energy, excluding and including depletion of the distribution due to annihilation (dashed and solid lines, respectively).

4. Summary and outlook

The Fortran program `ANTICOOL`, designed for the study of positron cooling via binary elastic collisions and annihilation in atomic gases, has been presented. Given user supplied scattering phase shifts, annihilation rates and annihilation γ spectra for the positron-atom system of interest, `ANTICOOL` enables the calculation of the time evolution of the positron momentum distribution $f(k,t)$, the fraction of positrons annihilating in a given time (lifetime spectrum), and the time-varying annihilation rate and γ spectra for positrons cooling in a thermal atomic gas. The effects of varying initial distributions and gas temperature can be studied. Its basic functionality and capabilities were demonstrated

using tutorial-style examples of positron cooling and annihilation in room-temperature helium and krypton gases.

Although beyond the scope of the current release, future developments should include (i) the introduction of positional-dependence so that the positron position and momentum are updated under a specified equation of motion, enabling the inclusion of user specified trapping potentials or external fields; and (iii) the ability to calculate positron cooling in molecules, i.e., account for additional inelastic processes such as rotational and vibrational excitations; (ii) the capability to study positron cooling in binary mixtures of atomic and molecular gases, important to study the effects of impurities.

Acknowledgements

DGG thanks Chris Harvey for valuable discussions. DGG is supported by the UK EPSRC (Engineering and Physical Research Council), grant EP/N007948/1.

Appendix A. Format of input data files

The user should must supply input data for scattering phase shifts. If an inelastic channel (e.g., annihilation for the positron) is present then one must also supply the inelastic cross section (for positron annihilation in He, Ne, Ar, Kr and Xe, the annihilation data is hard-coded). An optional input data file is the annihilation γ-spectrum $w_k(\epsilon)$, which is a function of both the momentum k and the Doppler-energy-shift ϵ. It is read if the flag is set to `.TRUE.` in '`collater.ini`'. The format of the files are described below.

If the user is to perform calculations on a general atom, or wishes to supply their own scattering phase shift or $Z_{\rm eff}(k)$ data, then they should set `atom="at"` in the `collater.ini` input file, and add the following lines to the `&SIMPRM` namelist:

- `alphad = X`,
  where `X` is the dipole-polarizability (in a.u.);
- `ascat = Y`,
  where `Y` is the scattering length (in a.u.);
- `m_at=Z`,
  where `Z` is the atomic mass (in amu).

Appendix A.1. Scattering phase shift input data

Table A.3: The variables in the input data file containing the elastic scattering phase shifts δ_ℓ: set as '`phasefile`' in `poscool.ini`.

! # comments	
3	
10	
57	
k_1 (a.u.)	$\delta_0(K_1)$ $\delta_1(K_1)$ $\delta_2(K_1)\ldots$
$\vdots$	$\vdots$
k_N (a.u.)	$\delta_0(K_N)$ $\delta_1(K_N)$ $\delta_2(K_N)\ldots$

The format of the scattering phase data input file is shown in Table A.3 (see also the file '`He_pha_spd.dat`' in the `.../anticool/example_He/` directory). The header line contains comments. The second line should specify the number of partial waves that phase shifts are supplied for in the file (in this case 3: those for s, p and d-wave positrons). The third line specifies the maximum number of partial waves to be included in the calculations: if this is larger than those supplied in the input file, then the remaining phase shifts are calculated using the O'Malley-Rosenberg-Spruch formula [27] and used in the calculation of the momentum-transfer cross section. The differential cross section assumes *all* δ_ℓ not supplied in the file are of O'Malley-Rosenberg-Spruch form [27], i.e., the sum to infinity of O'Malley-Rosenberg-Spruch phaseshifts is performed [see Eqn. (31) of [25]]. The third line specifies the number of momentum values at which the phase shifts being supplied are given for. The next lines contain the values of the scattering phase shifts at those k.

Appendix A.2. normalised annihilation rate Zeff)

If the user wishes to supply their own $Z_{\rm eff}(k)$ data, then they should be supplied in the Padé-approximant form:

$$Z_{\rm eff}(k) = \frac{\sum_{n=0}^{N} a_n k^n}{\kappa^2 + k^2 + \sum_{m=3}^{M} b_{m-3} k^m}, \qquad \text{(A.1)}$$

where $\kappa = 1/a$ is the reciprocal of the scattering length, b_m and c_m are constant coefficients determined from a fit to the true $Z_{\rm eff}$, and terms up to $M, N \leq 7$ are included to improve the accuracy of the fit. The values of b and c can be supplied to the program by simply modifying the '`poscool.ini`' input file by adding the Padé coefficients to the namelist:

- `a0=X,`
  ⋮
  `a7=X,`
  `b0=X,`
  ⋮
  `b4=X,`

(see '`mod_readinput.f90`' and '`mod_zeffpade.f90`' for the Padé coefficients for He, Ne, Ar, Kr and Xe.

Appendix A.3. Positron-momentum-dependent γ-spectra input data file

The format of the γ-spectra data input file is shown in Table A.4 (see also the file '`gamspec_He.dat`' in the `.../anticool/example_He/` directory). The first line contains the number of Doppler-energy-shift points the γ spectra specified in the file (in this case 501: as the spectra are symmetric, only the positive Doppler-energy-shift side is specified). The second line states the number of positron momenta for which the γ-spectra is specified for in the file. The next lines state those many values of the positron momenta (in a.u.). The subsequent lines give the Doppler-energy shift (in keV) and the corresponding γ spectrum point (in a.u.) for each momentum (spanning the columns).

Table A.4: Format of the γ spectra $w_k(\epsilon)$ input data file '`gamspec_atom.dat`'.

501
57
k_1 (a.u.)
⋮
k_N (a.u.)
ϵ_1 $w_{k_1}(\epsilon_1)$ $w_{k_2}(\epsilon_1)$ $w_{k_3}(\epsilon_1)$ ...
⋮
ϵ_i $w_{k_1}(\epsilon_i)$ $w_{k_2}(\epsilon_i)$ $w_{k_3}(\epsilon_i)$ ...
⋮

[1] J. R. Danielson, D. H. E. Dubin, R. G. Greaves, and C. M. Surko, Rev. Mod. Phys. **87**, 247 (2015).
[2] C. Hugenschmidt, Surf. Sci. Rep. **71**, 547 (2016).
[3] C. M. Surko, G. F. Gribakin, and S. J. Buckman, J. Phys. B **38**, R57 (2005).
[4] T. C. Griffith and G. R. Heyland, Phys. Rep. **39**, 169 (1978).
[5] M. Charlton, Rep. Prog. Phys. **48**, 737 (1985).
[6] D. G. Green, Phys. Rev. Lett. (accepted), arXiv:1706.01434.
[7] H. Stoll, K. M. M. Koch, and J. Major, Nucl. Instrum. and Meth. B **56**, 582 (1991).
[8] *Positron beams and their applications*, P. Coleman, ed., (World Scientific, 2000).
[9] A. Siegle, H. Stoll, P. Castellaz, J. Major, H. Schneider, and A. Seeger, App. Surf. Sci. **116**, 140 (1997).
[10] J. Engbrecht, Nuc. Instrum. Meth. B **221**, 119 (2004).
[11] U. Ackermann, B. Löwe, M. Dickmann, J. Mitteneder, P. Sperr, W. Egger, M. Reiner, and G. Dollinger, New J. Phys. **18**, 113030 (2016).
[12] D. G. Green, Phys. Rev. Lett. (accepted), arxiv.org/abs/1706.04948.
[13] I. Al-Qaradawi, M. Charlton, I. Borozan, R. Whitehead, and I. Borozan, J. Phys. B **33**, 2725 (2000).
[14] M. R. Natisin, J. R. Danielson, and C. M. Surko, J. Phys. B **47**, 225209 (2014).
[15] M. R. Natisin, J. R. Danielson, and C. M. Surko, Appl. Phys. Lett. **108**, 024102 (2016).
[16] E. M. Lifshitz and L. P. Pitaevskii, *Physical Kinetics - Course of Theoretical Physics Volume 10* (Pergamon, Oxford, 1981).
[17] P. H. R. Orth and G. Jones, Phys. Rev. **183**, 16 (1969).
[18] R. I. Campeanu and J. W. Humberston, J. Phys. B **10**, 239 (1977).
[19] R. I. Campeanu, J. Phys. B **14**, L157 (1981).

[20] R. I. Campeanu, Can. J. Phys. **60**, 615 (1982).
[21] B. Shizgal and K. Ness, J. Phys. B **20**, 847 (1987).
[22] G. J. Boyle, M. J. E. Casey, R. D. White, and J. Mitroy, Phys. Rev. A **89**, 022712 (2014).
[23] A. Farazdel and I. R. Epstein, Phys. Rev. A **16**, 518 (1977).
[24] P. Mullan, M. Lee, D. G. Green, and G. F. Gribakin, "Fokker-planck calculations of positron cooling and annihilation in noble gases," To be submitted to Phys. Rev. A.
[25] D. G. Green, J. A. Ludlow, and G. F. Gribakin, Phys. Rev. A **90**, 032712 (2014).
[26] L. D. Landau and E. M. Lifshitz, *Quantum Mechanics (Non-relativistic Theory) - Third Edition - Course of Theoretical Physics Volume 3* (Pergamon, Oxford, 1977).
[27] T. F. O'Malley, L. Spruch, and L. Rosenberg, J. Math. Phys. **2**, 491 (1961).
[28] M. K. Ali and P. A. Fraser, J. Phys. B **10**, 3091 (1977).
[29] P. A. Fraser, Adv. At. Mol. Phys. **4**, 63 (1968).
[30] I. Pomeranchuk, Zh. Eksp. Teor. Fiz. **19**, 183 (1949).
[31] D. G. Green and G. F. Gribakin, Phys. Rev. Lett. **114**, 093201 (2015).
[32] V. B. Berestetskii, E. M. Lifshitz, and L. P. Pitaevskii, *Quantum Electrodynamics*, 2nd ed. (Pergamon, Oxford, 1982).
[33] L. J. M. Dunlop and G. F. Gribakin, J. Phys. B **39**, 1647 (2006).
[34] Y. Sano, Y. Kino, T. Oka, and T. Sekine, Jpn. J. Appl. Phys. Conf. Proc. **2**, 011004 (2014).
[35] "Videos at http://www.am.qub.ac.uk/users/dgreen09/coolingvideos.html," .